

Pythonプログラミング入門

第1節 演算子と画面表示

画面に表示するときのポイントは、数値と文字を混在させないことです。

```
In [ ]: #演算子
print(1 + 1)
print(4 - 1)
print(4 * 2)
print(7 / 2)
print(7 // 2)
print(7 % 2)
print(3 ** 2)
print("Hello World!")
print("Hello " + "World!")
print('Hello World!')
print("Hello" * 3)
#16 + "Years" #数値と文字を足すとエラーになります。
print(str(16) + "Years")
```

第2節 コメント（注釈）の挿入方法とstr関数

```
In [ ]: """
複数行のコメント
このように書きます
"""

print("令和"+str(1)+"年")
print("西暦"+str(1+2018)+"年")

print("令和"+str(1)+"年", end="")
print("西暦"+str(1+2018)+"年")

print("""
こんにちは
おはよう
こんばんは
""")
```

第3節 変数とキーボード入力

Pythonでは変数に値を代入した時点で型を類推してくれます。

```
In [ ]: user_name = "Takeshi Izawa"
age = 42
next_age = age + 1
print(age)
print(next_age)
print(user_name)
year = input("今年は何年ですか?")
print(year)
year = int(year) + 1
print("来年は" + str(year) + "年です")
```

第4節 type関数,id関数,リスト,タプル

リストは可変長の配列、タプルは配列であるが、変更ができない

```
In [ ]: x = 5
y = 3.4
z = "Hello"
print(type(x))
print(type(y))
print(type(z))
z = "Hello ¥" Takeshi"
print(z)
z = "Hello ¥n Takeshi"
print(z)
print(float("3.123"))
print(int("4"))
print(str(3.445))
print(str(5))
name = ["Takeshi", "Hiroko", "Aiko"]
print(name[0])
name[0] = "Kazuya"
print(name)
print(len(name))
season = "Spring", "Summer", "Fall", "Winter"
weekday = ("Sunday", "Monday")
print(season)
print(season[0])
print(weekday)
print(id(weekday))
print(id(season))
```

第5節 モジュールのインポートと関数の利用

Pythonにはプログラミングを簡単にするための様々なAPIが用意されています。

```
In [ ]: #モジュールのインポート
import math
from calendar import TextCalendar
#インスタンスの作成
cal = TextCalendar()
#メソッドの呼び出し
cal.prmonth(2018, 2)
#引数を指定してコンストラクタを呼び出し
cal = TextCalendar(6)
#メソッドの呼び出し
cal.prmonth(2018, 2)

cals = cal.formatmonth(2018, 3)
print(cals)
num = math.sqrt(81)
print(num)
#piを表示
print(math.pi)
#乱数の利用
import random
print(random.randint(15, 20))
print(random.randint(15, 20))
print(random.randint(15, 20))
print(random.randrange(5))
```

第6節 if文

```
In [ ]: kazu = int(input("数を入力してください:"))
print("偶数" if kazu % 2 == 0 else "奇数")
tokuten = int(input("Please input Score:"))
if tokuten >= 80:
    print("A")
elif tokuten >= 70:
    print("B")
elif tokuten >= 60:
    print("C")
else :
    print("D")
```

第7節 ループ

```
In [ ]: #for文 イテレート可能なオブジェクトを利用したループ
seibetsu = ['男性', '女性']
for sei in seibetsu:
    print(sei)
#for文 rangeオブジェクトを使ったループ
for cnt in range(2):
    print(cnt)
#for文 rangeオブジェクトを使ったループ 1から始まって5未満の間、1ずつカウントアップ
for cnt in range(1,5,1):
    print(cnt)
#rangeオブジェクトを使ってリストを作成
guusu = list(range(0, 20, 2))
print(guusu)
#while文
kazu = 10
cnt = 0
goukei = 0
while cnt <= kazu:
    goukei = goukei + cnt
    cnt = cnt + 2
    print(goukei)
print(goukei)
```

第8節 break文,continue文,enumerate関数,zip関数

```
In [ ]: #中国地方の県
kens = ["Tottori", "Shimane", "END", "Hiroshima", "Okayama", "Yamaguchi"]
#中国地方の県庁所在地
citys = ["Tottori", "Matsue", "Hiroshima", "Okayama", "Yamaguchi"]

#break文の利用
for ken in kens:
    if ken == "END":
        print("Break LOOP")
        break
    print(ken)
#breakしたらelseは実行されない
else:
    print("End LOOP")

#continue文の利用
```

```

for ken in kens:
    if ken == "END":
        print("Continue LOOP")
        continue
    print(ken)
else:
    print("End LOOP")

#enumerate関数の利用
for index, ken in enumerate(kens):
    print(str(index) + ":" + ken)

#zip関数の利用
for (ken, city) in zip(kens, citys):
    print("Ken:" + ken + " City:" + city)

```

第9節 例外処理

```

In [ ]: import sys
try:
    age = int(input("年齢を入力してください"))
except:
    print("数値の入力ではなく文字が入力されました")
    sys.exit()
else:
    if age >= 18:
        print("成年です")
    else:
        print("未成年です")

```

第10節 内包表記

難易度 高 内包表記でコードを書くと高速にどうさせることができます。q

```

In [ ]: #2の0乗から2の16乗までをリストに格納
bin = []
for n in range(0, 17, 1):
    bin.append(2**n)
print(bin)
#内包表記で記述
bin = [2**n for n in range(0, 17, 1)]
print(bin)
#リストをもとに新しいリストを作成
price = [100, 250, 300]
tax = 1.08
total = [int(p*tax) for p in price]
#文字列を利用したリストの作成
rank = [r + "ランク" for r in "ABCDEF"]
print(rank)
#条件を満たす要素だけでリストを作成
guusuu = [n for n in range(0, 10, 1) if n % 2 == 0]
print(guusuu)
#指定した県の市の名前を表示
citys=["鳥取県米子市", "鳥取県境港市", "広島県福山市"]
tottori_citys = [city[3:] for city in citys if city.startswith("鳥取県")]
print(tottori_citys)
#合格者の抽出
test = [("A", "合格"), ("B", "合格"), ("C", "不合格"), ("D", "合格"), ]
passed = [x[0] for x in test if x[1] == "合格"]
print(passed)

```

```
#辞書の内包表記の利用
teams = ["A", "B"]
team_dic = {t:0 for t in teams}
print(team_dic)
points = [40, 59]
team_dic = {t:p for t,p in zip(teams, points)}
print(team_dic)
team_dic = {str(p):t for (t,p) in team_dic.items()}
print(team_dic)
#集合の内包表記
point = [10, 10, 20, 100, 40, 50, 60]
fails = {x for x in point if x <=60}
print(fails)
```

第11節 関数の宣言と利用

```
In [ ]: #税込み金額の価格を返す関数
def getPrice(price, tax):
    return int(price*tax)

print(getPrice(100, 1.08))
print(getPrice(price=100, tax=1.08))
#print(getPrice(price=100, 1.08)) #エラーになる
print(getPrice(100, tax=1.08))
print(getPrice(tax=1.08, price=100))
#引数にデフォルト値を設定する
def getPrice2(price, tax=1.08):
    return int(price*tax)
print(getPrice2(1000))
#リストの値を変更する
list = ["A", "B"]
def addC(list_item):
    list_item.append("C")

addC(list)
print(list)
#グローバル変数の利用
x = 0
def setCount(y):
    x = y
def setCount2(y):
    global x
    x = y
setCount(10)
print(x)
setCount2(12)
print(x)
```

第12節 関数で可変長引数を利用する, Lambda式の利用

```
In [ ]: #可変長引数を用いた関数
def keisan(*kazu, mode):
    total = 0
    if mode == "+":
        for i in kazu:
            total += i
    if mode == "-":
        for i in kazu:
```

```

        total -= i
    return total
print(keisan(1, 2, 3, 4, 5, mode="+"))
print(keisan(1, 2, 3, 4, 5, mode="-"))
#引数を辞書で受け取る
def insert_data(**data):
    print(data)
insert_data(name="Izawa", blood="A", age="43")
#Lambda式の利用
bigger = lambda kazu1, kazu2: kazu1 if kazu1 > kazu2 else kazu2
print(bigger(10, 4))
print(bigger(100, 400))
getPrice = lambda price, tax=1.08 : int(price*tax)
print(getPrice(1000))
print(getPrice(1000, 1.1))

```

第13節 map関数,無名関数

In []:

```

#map関数の利用
def getTaxPrice(price, tax=1.08):
    return int(price * tax)

prices = [100, 250, 300, 1400]
for price in map(getTaxPrice, prices):
    print(price)
#リストに保存
onTaxPrices = list(map(getTaxPrice, prices))
print(onTaxPrices)

#無名関数とmap関数
prices = [100, 250, 300, 1400]
onTaxPrices = list(map(lambda price, tax=1.08: int(price*tax), prices))
print(onTaxPrices)

#map関数の代わりに内包表記を利用
prices = [100, 250, 300, 1400]
onTaxPrices = [int(price * 1.08) for price in prices]
print(onTaxPrices)

#filter関数の利用
def over1000(price):
    return price >= 1000

prices = [100, 250, 300, 1400]
for price in filter(over1000, prices):
    print(price)

#無名関数とfilter関数
prices = [100, 250, 300, 1400]
for price in filter(lambda price: price >= 1000, prices):
    print(price)

#filter関数の代わりに内包表記を利用
prices = [100, 250, 300, 1400]
over1000Prices = [price for price in prices if price >= 1000]
print(over1000Prices)

#辞書のソート ソートするkeyを指定します。
teams = {"Zaku": "Ateam", "Bash": "Ateam", "Ken": "Bteam", "Take": "Ateam"}
print("名前の昇順")
for player in sorted(teams.items(), key=lambda k: k[0]):
    print(player[0], player[1])

```

```

print("チームの昇順")
for player in sorted(teams.items(), key=lambda k:k[1]):
    print(player[0], player[1])

#ジェネレータ関数の利用
def weekdays():
    wdays = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
    for w in wdays:
        yield w
w = weekdays()
print(next(w))
print(next(w))
print("一休み")
w = weekdays()
for x in w:
    print(x)

#ジェネレータ式の利用 (内包表記より高速に動作します)
wdays = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
w = (w for w in wdays)
for x in w:
    print(x)

```

第14節 クラスの宣言と利用

In []:

```

#クラスの作成
class TeamA:
    member = 0 #メンバーの人数 (クラス変数)
    def __init__(self, name, age):
        self.name = name #引数の値をインスタンス変数に代入
        self.age = age
        TeamA.member += 1 #クラス変数に1加算
    def getBornYear(self):
        return 2019 - self.age

izawa = TeamA("Takeshi", 43)
print(izawa.name+": "+str(izawa.age))
izawa.name = "Takeshi Izawa"
print(izawa.name)
yamada = TeamA("hanako", 23)
print("メンバーの人数:"+str(TeamA.member))
print(izawa.name+": "+str(izawa.getBornYear())+"年生まれ")

```